บุญเลิศ เอี่ยมทัศนา*

การ์ดเฮอร์คิวลิสกับภาษาไทย

การติดตั้งภาษาไทยให้แก่เครื่องคอมพิวเตอร์ โดยเฉพาะบนเครื่องไมโครคอมพิวเตอร์ได้ดำเนินการกันมานานแล้ว และก็รู้จักกันแพร่หลายในตลาด แต่ก็มีผู้สนใจไต่ถามอยู่เนือง ๆ ว่าทำได้อย่างไร ยากนักหรือ? คิดว่าผู้ถามคงมีความในใจเกี่ยวกับการจัดซื้อจัดหามากกว่าจะดำเนินการเอง บทความบทนี้จึงจะพยายามสนองตอบความต้องการดังกล่าว และโดยเฉพาะทางด้านฮาร์ดแวร์

โดยที่การแสดงภาพอักขระบนจอภาพสามารถทำได้ 2 แบบ คือ คาแรกเตอร์โหมด (charactermode) ซึ่งมาถึงสมัยนี้นิยมเรียกว่า เท็กซ์โหมด (text mode) กับแบบกราฟิกโหมด (graphics mode) การที่จะให้เครื่องแสดงภาพอักษรไทยจึงทำได้ 2 โหมดเช่นกัน แต่ผมเห็นว่า ถ้าของเดิมเขาทำในโหมดใด เราก็ควรทำในโหมดนั้น

สำหรับเครื่องไมโครคอมพิวเตอร์ที่แพร่หลายในบ้านเราในปัจจุบันได้แก่ แอปเปิ้ล ทู กับ ไอบีเอ็ม พีซี/เอ็กซ์ที ทั้งของแท้และของเทียม ทั้งคู่แสดงภาพอักขระในแบบคาแรกเตอร์โหมด ทั้งนี้เพราะในกราฟิกโหมด เราอาจจะทำให้ภาพอักขระคมชัดกว่า แต่ก็ช้ากว่าแบบเท็กซ์โหมด สำหรับของเครื่องแอปเปิ้ลผมได้กล่าวไว้แล้วในบทความเรื่อง *"สอนแอปเปิ้ลเขียนไทย"* ในไมโครฯ ฉบับที่ 14 ถึง 16 ในฉบับนี้ ผมจึงจะนำเรื่องราวของไอบีเอ็มมากล่าว และที่เลือกพูด

* รองหัวหน้าฝ่ายวิชาการสถาบันวิชาการทหารเรือชั้นสูง (ส.ร.ส.)

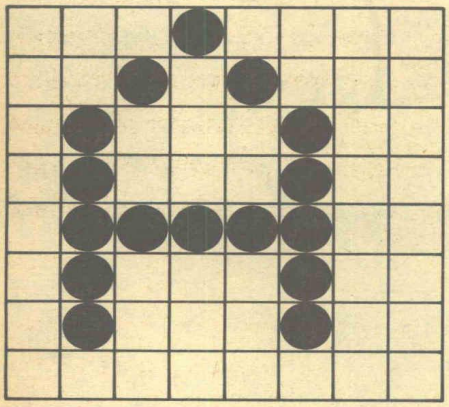
เฉพาะการ์ดเฮอร์คิวลิสเพราะรู้จักกันมากและสะดวกที่จะทำภาษาไทย 25 บรรทัดด้วย

การแสดงภาพอักขระ

ก่อนที่จะติดตั้งภาษาไทยให้แก่เครื่องซึ่งในกรณีนี้ ผมหมายถึง จะให้เครื่องแสดงภาพอักษรไทยได้อย่างไรนั้น จำเป็นต้องทราบว่าเครื่องเดิมเขาแสดงภาพอักขระได้อย่างไร ซึ่งจะหมายถึงในเท็กซ์โหมด และในบทความนี้จะหมายถึงเท็กซ์โหมดเท่านั้น มีเรื่องราวหลายเรื่องที่จะต้องพูดถึง เพื่อให้สั้นผมจะกล่าวถึงแต่ละเรื่องคล้ายแนะนำตัวละคร แล้วถึงจะนำมารวมกัน ซึ่งจะเป็นเรื่องราวของการแสดงภาพอักขระโดยทั่วไป ไม่ใช่ของไอบีเอ็มโดยตรงก่อน จากนั้นจะพูดถึงการ์ดของไอบีเอ็ม

การแสดงภาพทางดิจิตอล

การแสดงภาพทางดิจิตอลมีแต่จุดขาวกับดำและเป็นจุด ๆ ไปไม่ต่อเนื่องเหมือนทางลิเนียร์ เช่น เครื่องรับทีวี แต่มีส่วนที่เหมือนกันคือแสดงเป็นสแกนไลน์เช่นกัน เช่นในเครื่องรับทีวีที่กล่าวว่าเป็น 625 เส้นนั้น จะมีเส้นขีดขวางจออยู่ 625 เส้น (ความจริงไม่ถึง เพราะจะต้องถูกแบ่งไปเป็นช่วงรีเทรซ (retrace) บางส่วน) เส้นต่าง ๆ เหล่านี้ โดยเฉพาะ **ในทีวีขาวดำเปรียบเสมือนการลากเส้นที่มีความหนักเบาไม่เท่ากันหลาย ๆ เส้น** ทำให้เรามองเห็นเป็นภาพ



รูปที่ 1 ภาพอักขระในเมตริกซ์ 8×8 ที่ต้องการแสดงออกจอภาพ

แต่ทางดิจิตอลเปรียบเสมือนเราจุด ๆ ไปตามแนวไม้บรรทัดไม่ใช่ลากเส้น ทำให้เกิดจุดภาพบนจอภาพ และจุดภาพนี้มีอยู่ 2 สถานะเท่านั้นคือ ขาวกับดำ ถ้าให้เป็นขาว จะปรากฏเป็นจุดขาวบนจอภาพ แต่ถ้าให้เป็นดำจะกลมกลืนกับพื้นหลังของจอภาพ ไม่ปรากฏเป็นจุดบนจอภาพ คือถ้าให้เป็นดำทั้งหมดก็จะไม่เห็นเส้นสแกนไลน์ ถ้าให้ขาวทั้งหมดจะเห็นเป็นเส้นประ

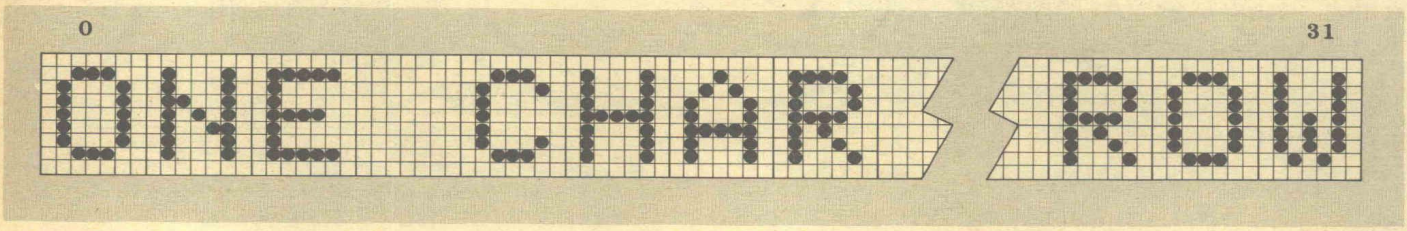
เส้นสแกนไลน์ 1 เส้น ไม่ทำให้เกิดเป็นภาพอักขระได้ เช่น เราจะให้แสดงภาพอักขระในเมตริกซ์ 8×8 ดังรูปที่ 1 จะต้องใช้เส้นสแกนไลน์ 8 เส้น เรียกว่า หนึ่งบรรทัดอักขระ (character row) เพราะจะแสดงตัวอักษรได้อีกหลาย ๆ ตัวดังแสดงในรูปที่ 2 ซึ่งในรูปที่ 2 เป็นการแสดงแบบ 32 อักขระต่อบรรทัด ใช้ในสมัยที่คุณภาพของจอภาพด้อยกว่าปัจจุบันมาก

ปัจจุบันคุณภาพของจอภาพทำให้เราสามารถแสดงได้ถึง 80 อักขระต่อจอภาพ และได้ถึง 25 บรรทัดอักขระต่อจอภาพ (ความจริงทำได้มากกว่านี้อีก) แต่ทั้งนี้ขึ้นอยู่กับเราจะกำหนด เช่น ถ้าต้องการให้แสดงในแบบเมตริกซ์ 8×9 จำนวน 64 อักขระต่อ 1 บรรทัดอักขระ และ 20 บรรทัดอักขระต่อ 1 จอภาพ ก็จะกำหนดให้มีจำนวนจุดภาพ 8×64 = 512 จุดต่อ 1 สแกนไลน์ และให้มี 9×20 เท่ากับ 180 สแกนไลน์ต่อ 1 จอภาพ ดังจะกล่าวในหัวข้อ CRTC แต่ทั้งนี้ต้องขึ้นกับความถี่ของคริสตอลที่นำมาทำเป็นสัญญาณจุดภาพด้วย

เมตริกซ์หนึ่ง ๆ ดังที่กล่าวมาไม่ว่าจะเป็น 8×8 หรือ 8×9 หรือจะเป็นขนาดอื่นอีก เรียกว่า 1 ช่องอักขระ สำหรับคำว่าบรรทัดหรือบรรทัดอักขระขอให้ถือว่ามีความหมายเหมือนกันเช่นเดียวกับคำว่า row และ character row โดยทั่วไปจะใช้คำว่า โรว์ คู่กับ คอลัมน์ เพื่อแสดงถึงตำบลที่ของอักขระบนจอภาพ คำว่าคอลัมน์ (column) หมายถึง ช่องอักขระเมื่อนับในแกนนอน เช่น คอลัมน์ 0 คืออักขระตัวแรกของบรรทัด คอลัมน์ 79 คือตัวสุดท้ายของบรรทัดเมื่อให้แสดงในแบบ 80 อักขระต่อบรรทัด

รหัสอักขระ

เรื่องต่อไปที่ผมขอนำมากล่าวถึงก่อนคือ รหัสอักขระบนเครื่องไมโครคอมพิวเตอร์เกือบทั้งหมดใช้แบบ ASCII ซึ่งย่อมาจาก American Standard Code for Information Interchangable ความหมายก็ไม่มีอะไรมากไปกว่าการกำหนดหมายเลขให้แก่อักขระต่าง ๆ ให้เป็นมาตรฐานเดียวกันเช่น วรรค (space) ให้มีรหัสเป็น 32 อักษร A เป็น 65 และอื่น ๆ รวมทั้งรหัสในการควบคุมด้วย



รูปที่ 2 ภาพอักขระที่ปรากฏบนจอภาพใน 1 บรรทัดอักขระ

ในภาษาอังกฤษ อักษรทั้งเล็กและใหญ่ ตัวเลขและเครื่องหมายต่าง ๆ รวมทั้งรหัสในการควบคุมอีก 32 รหัส รวมกันแล้วไม่ถึง 128 รหัส แต่ได้เพิ่มเครื่องหมายต่าง ๆ จนครบ มีรหัสเป็น 0 ถึง 127 หรือ $00 ถึง $7F (เครื่องหมาย $ หมายความว่า ตัวเลขที่ตามมาเป็นเลขฐานสิบหก) ดังนั้นจึงใช้หน่วยความจำเพียง 7 บิทใน 1 ไบท์เท่านั้น ทำให้รหัสอักขระในภาษาอังกฤษเป็นรหัสจอภาพไปในตัว ซึ่งผมขอพักเรื่องนี้ไว้เพียงเท่านี้ก่อน

แต่มีเรื่องที่ขอกล่าวเสริมว่า เช่น ในหน่วยความจำมีข้อมูลเป็น 0100 0001 เป็นเลขฐานสอง หรือคือ 65 ในเลขฐานสิบ หากเรามองเลขจำนวนนี้เป็นเลขจำนวนก็จะมีค่าเป็น 65 แต่ถ้าเรามองเป็นรหัสอักขระก็จะเป็นอักษร A การมองนี้ ผู้เขียนโปรแกรมเป็นผู้มองครับ เพราะเกี่ยวกับที่เขาจะนำไปดำเนินกรรมวิธีต่อไป แต่ต้องเป็นผู้เขียนโปรแกรมในระดับภาษาเครื่องถึงจะมองเห็นถนัด

หน่วยความจำจอภาพ
(video RAM หรือ screen memory)

ในการแสดงภาพอักขระทางจอภาพที่นิยมกันในไมโครคอมพิวเตอร์คือ จัดหน่วยความจำหน่วยหนึ่งเป็นหน่วยความจำจอภาพ โดยที่ต้องจัดวงจรทางฮาร์ดแวร์ไว้ว่า ถ้าหากนำค่ารหัสอักขระใดก็ตามมาไว้ที่ไบท์ศูนย์ของหน่วยความจำนี้ จะปรากฏภาพอักขระนั้น บนจอภาพตรงมุมบนซ้าย ถ้าไว้ที่ไบท์ 1 ก็จะปรากฏเป็นตัวที่สองในบรรทัดเดียวกัน เรียงกันไปถึงไบท์ 79 ถ้าเป็นจอภาพแบบ 80 คอลัมน์ ก็จะเป็นตัวสุดท้ายของบรรทัดนั้น ไบท์ 80 จะเป็นตัวแรกของบรรทัดใหม่ เรียงกันไปเช่นนี้จนถึงไบท์สุดท้ายจะปรากฏภาพที่มุมล่างขวา

ขนาดของหน่วยความจำจอภาพสามารถคำนวณได้ เช่น ต้องการให้แสดงภาพในแบบ 80 อักขระต่อบรรทัด และแสดง 25 บรรทัด ขนาดของหน่วยความจำจอภาพจะเป็น 80×25 = 2,000 ไบท์ ผมหมายถึงโดยทั่วไป สำหรับของไอบีเอ็มมีเรื่องราวพิเศษซึ่งจะกล่าวถึงต่อไป สำหรับหน่วยความจำจอภาพนี้ เราจะให้อยู่ที่แอดเดรสใดก็ได้ ขึ้นอยู่กับการออกแบบ

มีเรื่องที่ขอให้สังเกตไว้ด้วยครับว่า รหัสอักขระ 1 รหัสคือ 1 ไบท์ ทำให้เกิดภาพอักขระ 1 ตัว เช่น ตามตัวอย่างที่ยกมาคือให้เป็นแบบเมตริกซ์ 8×8 จุดภาพ หากให้เป็นจุดขาวทั้งหมด (บางเครื่องให้เป็นรหัส $A0) ก็จะบังเกิดเป็นจุดภาพทั้งหมด 64 จุดภาพบนจอภาพ

มีเรื่องที่ผมไม่ได้กล่าวมาแต่ต้นว่า การทำให้บังเกิดจุดภาพ 1 จุดจะต้องใช้หน่วยความจำ 1 บิท และส่วนมากจะให้เป็นจุดขาวเมื่อบิทนี้มีค่าเป็น 1 และเป็นจุดดำเมื่อมีค่าเป็น 0 ดังนั้นการสร้างภาพ 8×8 จุดภาพ จะต้องใช้หน่วยความจำ 8 ไบท์ หน่วยความจำที่จะใช้ทั้งหมดจะต้องเป็น 80×25×8 = 16,000 ไบท์

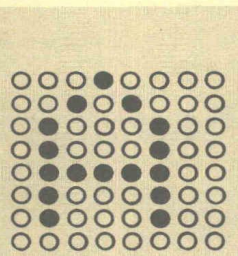
แต่ทำไมผมจึงกล่าวว่าใช้หน่วยความจำเพียง 2,000 ไบท์ ทั้งนี้เพราะเขาใช้คาแรกเตอร์เยนฯ (character generator) เข้าช่วย และถ้าไม่ใช้คาแรกเตอร์เยนฯ คือจัดหน่วยความจำเต็ม 16,000 ไบท์ ก็คือเป็นการแสดงภาพในแบบกราฟิกโหมดครับ

คาแรกเตอร์เยนฯ
(character generator)

จากที่ผมกล่าวแล้วว่าเราใช้คาแรกเตอร์เยนฯ เข้าช่วยในการแสดงภาพอักขระในแบบคาแรกเตอร์โหมด คาแรกเตอร์เยนฯ คือหน่วยความจำประเภท ROM ปัจจุบันที่นิยมใช้กันมากคือ EPROM ซึ่งถูกโปรแกรมแพทเทอร์นของภาพอักขระไว้ ขอให้ดูในรูปที่ 3 เป็นการเตรียมข้อมูลเพื่อโปรแกรม EPROM สำหรับการแสดงผลในแบบเมตริกซ์ 8×8 โดยสมมติว่าได้ดำเนินการมาถึงอักษร A

ในการแสดงผลแบบ 8×8 เราต้องการหน่วยความจำ 8 ไบท์ต่อแพทเทอร์นของภาพอักขระ 1 อักขระ ดังนั้นหากต้องการแสดงภาพอักขระทั้ง 256 อักขระ ต้องใช้หน่วยความจำ 256×8 = 2,048 หรือคือ 2 K ไบท์ ได้แก่ EPROM เบอร์ 2716 การโปรแกรม EPROM เราจะโปรแกรมให้รหัส $00 อยู่ที่ไบท์ 0 ถึงไบท์ 7 และรหัส $01 อยู่ที่ไบท์ 8 ถึงไบท์ 15 และต่อ ๆ ไป ดังนั้นรหัสของ A คือ $41 หรือ 65 จึงอยู่ที่ไบท์ 520 ถึงไบท์ 527

จากที่กล่าวมานี้จะเห็นว่าแพทเทอร์นของภาพอักขระเก็บอยู่ในลักษณะเป็นไบท์คูณ 8 แต่การที่เราจะให้ค่าในแต่ละไบท์แก่จอภาพได้นั้น เราต้องจับมันมาเข้าแถวเรียงเดี่ยวกันออกไป ซึ่งทำได้โดยใช้ชิฟท์รีจิสเตอร์



0	0	0	1	0	0	0	0	10H
0	0	1	0	1	0	0	0	28H
0	1	0	0	0	1	0	0	44H
0	1	0	0	0	1	0	0	44H
0	1	1	1	1	1	0	0	7CH
0	1	0	0	0	1	0	0	44H
0	1	0	0	0	1	0	0	44H
0	0	0	0	0	0	0	0	00H

รูปที่ 3 แพทเทิร์นของอักขระ A ช่องขวาสุดเป็นเลขฐานสิบหก

ชิฟท์รีจิสเตอร์ (shift register)

ชิฟท์รีจิสเตอร์ทำหน้าที่เปลี่ยนข้อมูลจากขนานไปเป็นอนุกรม เพื่อจะนำข้อมูลจากคาแรกเตอร์เยนฯ ไปออกจอภาพ ซึ่งจะดำเนินการครั้งละ 1 ไบท์ ได้ภาพบนจอภาพ 8 จุดภาพ ดังแสดงในรูปที่ 4

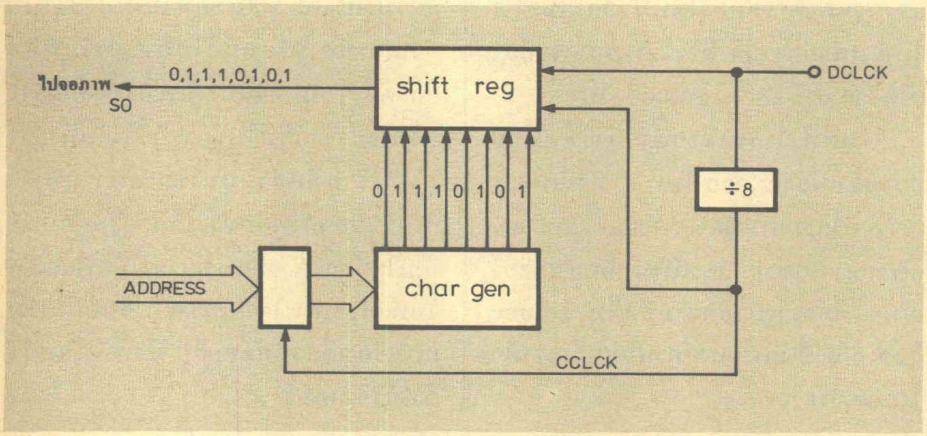
ในรูปแสดงว่ามีอุปกรณ์ให้ค่าแอดเดรสแก่คาแรกเตอร์เยนฯ ตามช่วงจังหวะของสัญญาณนาฬิกา C CLCK (character clock) ซึ่งคาแรกเตอร์เยนฯ จะให้ค่าดาต้าแก่ชิฟท์รีจิสเตอร์ นำไปชิฟท์ออกเป็นข้อมูลแบบอันดับตามช่วงจังหวะของสัญญาณนาฬิกา D CLCK (dot clock) ซึ่งเร็วกว่า C CLCK แปดเท่า

ที่คั่นไว้ด้วยเครื่องหมาย , เพื่อแสดงว่าค่าที่ปรากฏบนสาย SO นี้ จะมีเพียงครั้งละ 1 ค่าเท่านั้น ตามช่วงจังหวะของ D CLCK คือจะมีค่าเป็น 0 แล้วเป็น 1 แล้วเป็น 1 ต่อ ๆ ไปจน D CLCK สุดท้ายมีค่าเป็น 1

เริ่มต้นรอบใหม่คือ อ่านข้อมูลจากคาแรกเตอร์เยนฯ ตามจังหวะของ C CLCK มาชิฟท์ออกตามช่วงจังหวะของ D CLCK วนเขียนอยู่เช่นนี้ ได้เป็นสายของตัวเลขฐานสอง 1/0 หรือลอจิก H/L อย่างต่อเนื่อง ให้แก่จอภาพทำให้บังเกิดเป็นจุดภาพขาว/ดำ บนจอภาพ

วงจรพื้นฐาน

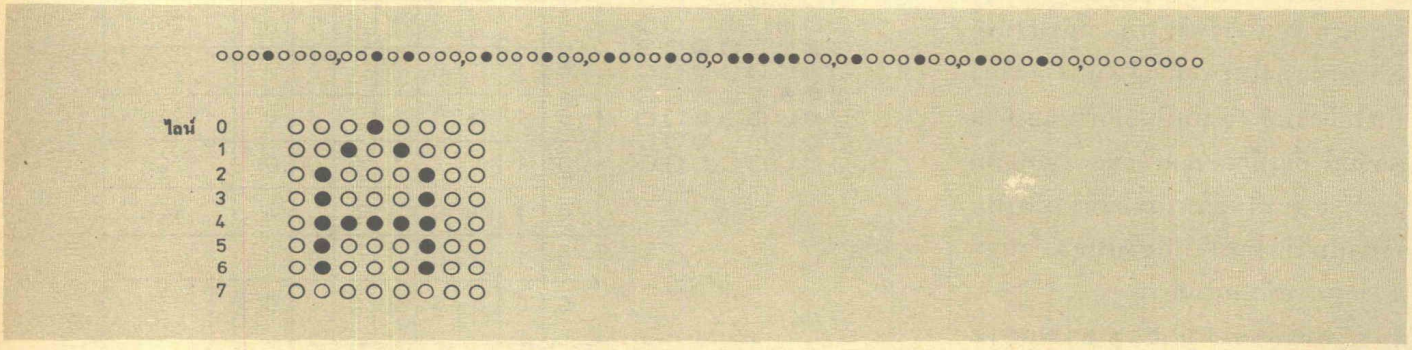
ถ้าเราจัดวงจรตามรูปที่ 4 นี้ และให้ค่าแอดเดรสแก่คาแรกเตอร์เยนฯ ซึ่งเป็น EPROM เบอร์ 2716 ตั้งแต่ 0 ถึง 2047 หรือตั้งแต่ $000 ถึง $7FF จะได้ภาพที่เปรอะไปหมดบนจอภาพ เช่น เมื่อถึงอักษร A จะได้ภาพดังรูป 5 ตอนบน แต่เราต้องการให้เป็นดังรูป 5 ตอนล่าง



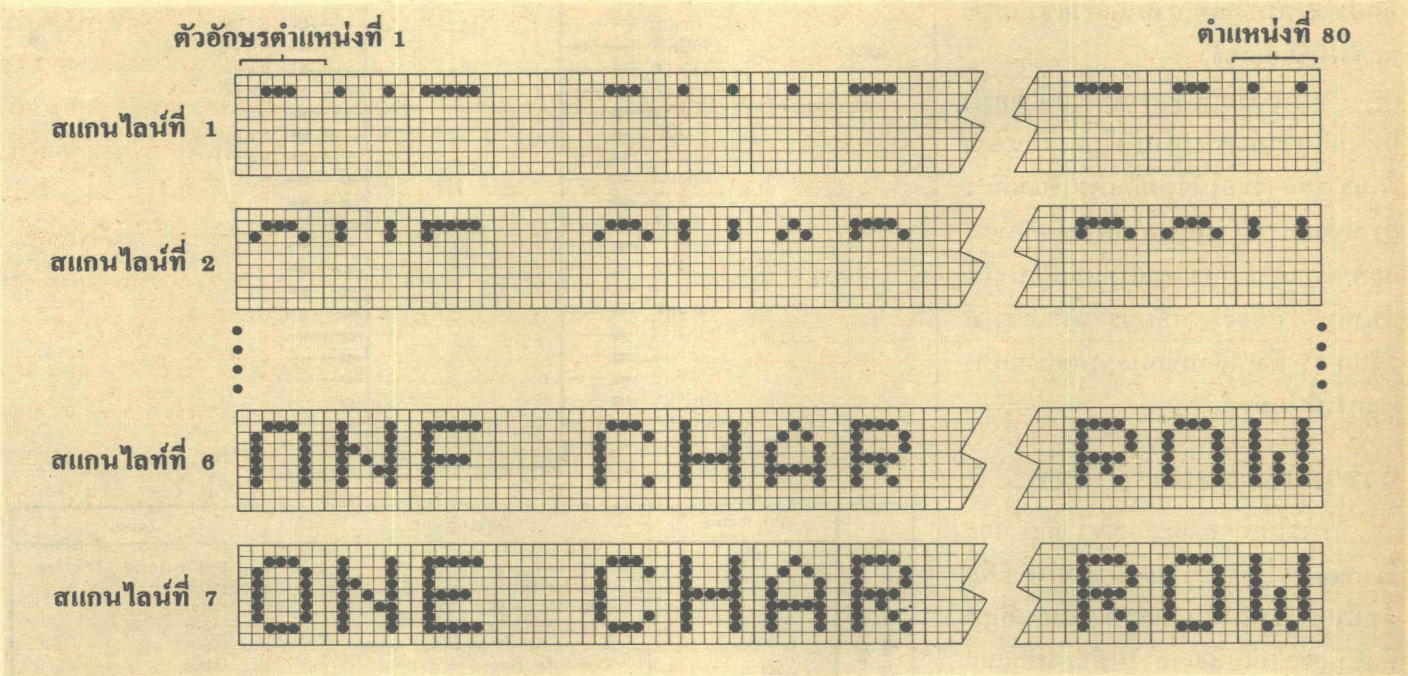
รูปที่ 4 แสดงถึงการทำงานของชิฟท์รีจิสเตอร์ กับคาแรกเตอร์เยนฯ

วิธีแก้ก็คือ จากตัวอย่างที่ยกมานี้เป็นการออกจอภาพในแบบเมตริกซ์ 8×8 ซึ่งใน 1 บรรทัดอักขระจะมี 8 สแกนไลน์ ดังที่ได้ใช้คำว่าไลน์กำกับไว้ในรูปที่ 5 ดังนั้นหากผมแยกขา A0 ถึง A2 ของคาแรกเตอร์เยนฯ รวมสามขาออกมาเรียกว่าเป็น L0 ถึง L2 คือขา lines ขาที่เหลืออีก 8 ขา คือ A3 ถึง A10 เรียกเสียใหม่ว่าเป็น C0 ถึง C7 คือขา characters จะเห็นว่าขา lines อ้างแอดเดรสได้ตั้งแต่ 0 ถึง 7 คือไลน์ 0 ถึง ไลน์ 7 ในแต่ละบรรทัดอักขระ ส่วนขา characters อ้างแอดเดรสได้ตั้งแต่ 0 ถึง 255 ก็คือรหัสอักขระของอักขระแต่ละตัวนั่นเอง

สมมติว่าผมต้องการออกจอภาพในแบบ 64 อักขระต่อบรรทัดและ 24 บรรทัดต่อจอภาพ ผมจะจัดวงจรใหม่ให้แอดเดรสแก่คาแรกเตอร์เยนฯ ดังนี้คือให้ขา lines มีค่าเป็น 0 ไป 64 C CLCK และให้ขา characters มีค่าเป็น 0 ถึง 63 โดยเปลี่ยนทุก C CLCK ผมจะได้ภาพ 1 สแกนไลน์บนจอภาพ คือภาพในไลน์ 0 ของอักขระ 0 ถึง 63 ซึ่งถ้าดูในตารางรหัส ASCII จะเป็นพวกรหัสควบคุมเสีย 32 อักขระ และเป็น space คือ วรรค, ! ไปจนถึง ? เสร็จแล้วผมให้ขา lines มีค่าเป็น 1 อีก 64 C CLCK ส่วนขา characters คงให้มีค่าเป็น 0 ถึง 63 ทำซ้ำอยู่เช่นนี้จน lines มีค่าเป็น 7 แล้วจะได้ภาพ 1 บรรทัดอักขระคือ โรว์ 0 เช่น



รูปที่ 5 ตอนบนแสดงภาพของอักษร A เมื่อยังไม่ดำเนินการในเรื่องสแกนไลน์ ตอนล่างเป็นภาพอักษร A ที่ต้องการให้ออกจอภาพ คือเมื่อได้ดำเนินการในเรื่องสแกนไลน์แล้ว



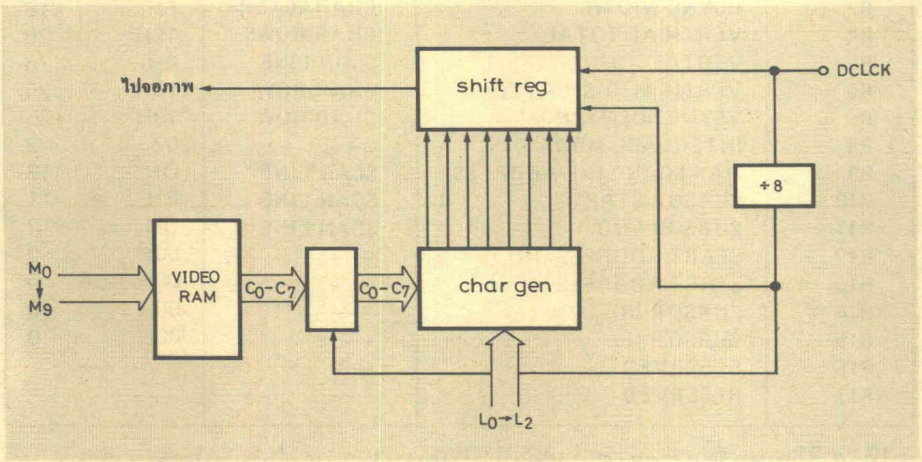
รูปที่ 6 แสดงการเกิดภาพบนจอภาพจาก line 0 ถึง line 7 ใน 1 บรรทัดอักขระ

เดียวกับรูปที่ 6 (แต่ภาพไม่เหมือนกัน)

เสร็จแล้วทำซ้ำเช่นนี้อีกในโรว์ 1 ถึงโรว์ 3 แต่ให้ค่าแก่ขา characters เป็น 64 ถึง 127 และ 128 ถึง 191 และ 192 ถึง 255 (คือ $40 ถึง $7F, $80 ถึง $BF และ $C0 ถึง $FF) ในโรว์ที่ 1, 2 และ 3 ตามลำดับ จะได้ภาพอักขระทุกตัวในคาแรกเตอร์เยนฯ บนจอภาพแต่ยังไม่ใช่สิ่งที่ต้องการ

จึงได้ปรับปรุงวงจรใหม่เป็นดังรูปที่ 7 โดยเพิ่มหน่วยความจำจอภาพ (video RAM) ตามตัวอย่างที่ยกมาคือ 64 อักขระต่อบรรทัด 25 บรรทัด หน่วยความจำหน่วยนี้จะมีขนาดเป็น 64×25 = 1,600 คือ $640 แต่เราต้องใช้หน่วยความจำขนาด 1 K ไบท์ ซึ่งมีขาแอดเดรส 10 ขา ได้ให้ชื่อเป็น M0 ถึง M9 คือขา memories

ผมจะต้องจัดวงจรให้ค่าแก่ขา lines เช่นเดิม ส่วนค่าที่จะให้แก่ขา characters ให้นำมาให้แก่ขา memories การดำเนินการคงเป็นเช่นเดิมคือ ในโรว์ 0 เริ่มให้ค่าขา memories ตั้งแต่ 0 ถึง 63 ซ้ำอยู่ 8 เที่ยวคือ ในไลท์ 0 ถึง 7 ในโรว์ 1 ให้ค่าขา memories ตั้งแต่ 64 ถึง 127 สำหรับโรว์ต่อ ๆ ไปจะเพิ่มค่าแรกขึ้นโรว์ละ 64 จนถึงโรว์ 24 ให้ค่าแก่ขา memories ตั้งแต่ 1536 ถึง 1599 ($600 ถึง $63F) แล้วไปขึ้นต้น 0 ใหม่ จะได้ภาพอักขระบนจอภาพทั้งหมด 1,600 ภาพ



รูปที่ 7 วงจรแสดงผลบนจอภาพเมื่อเพิ่มหน่วยความจำจอภาพ

ภาพอักขระที่ปรากฏบนจอภาพจะเป็นไปตามรหัสอักขระที่เราให้ไว้ในหน่วยความจำจอภาพ (การให้ค่ารหัสอักขระแก่หน่วยความจำจอภาพยังไม่ได้กล่าวถึงครับ) เช่น ให้รหัสอักขระ Z หรือ $5A ไว้ที่ไบท์ 0 ของหน่วยความจำจอภาพ ก็จะปรากฏภาพอักษร Z ที่มุมบนซ้ายของจอภาพ

ที่เป็นเช่นนี้เพราะเมื่อขา memories เป็น 0 หน่วยความจำจอภาพจะให้ค่า $5A ออกทางขาดาต้า กลายเป็นค่าขา characters ให้แก่ คาแรกเตอร์เยนฯ ซึ่งคาแรกเตอร์เยนฯ จะให้ค่าแพทเทอร์นของ

อักษร Z ตามไลน์ที่กำหนดด้วยขา lines แก่ชิฟท์รีจิสเตอร์

หากสงสัยก็กรุณาทบทวนด้วยครับ แต่ผมคิดว่าสำหรับท่านที่ไม่คุ้นเคยทางด้านฮาร์ดแวร์ขอให้จำหลักการไว้แต่เพียงว่า หน่วยความจำจอภาพให้ค่ารหัสอักขระแก่คาแรกเตอร์เยนฯ คาแรกเตอร์เยนฯให้แพทเทอร์นของภาพอักขระนั้นแก่ชิฟท์รีจิสเตอร์ ซึ่งเปลี่ยนข้อมูลจากขนานเป็นอนุกรมให้แก่จอภาพ

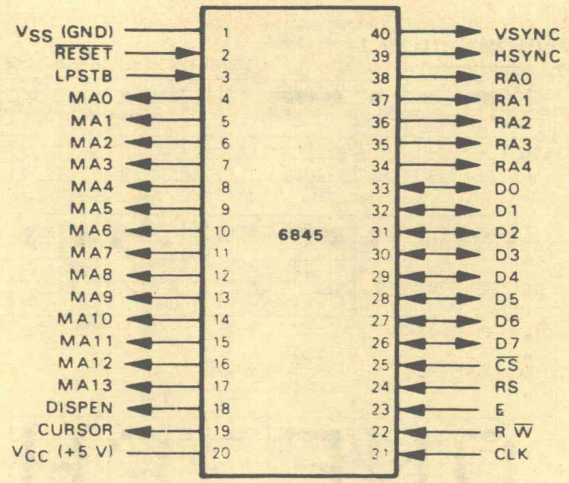
วงจรควบคุมจอภาพ CRTC

การให้ค่าแอดเดรสแก่ขา lines และ memories ดังกล่าวมานี้ หากให้ซีพียูดำเนินการจะไม่ทัน และที่แน่นอนที่สุดคือซีพียูจะไม่เหลือเวลาไปทำงานอื่นเลย จึงได้สร้างวงจรควบคุมจอภาพคือ CRTC (Cathode Ray Tube Controller) ขึ้นมาทำหน้าที่นี้ ที่รู้จักกันแพร่หลายคือเบอร์ 6845 ของโมโตโรลาซึ่งไอบีเอ็มได้นำมาใช้ในเครื่องพีซีด้วย

ลักษณะภายนอกของ 6845 เป็นไอซีขนาด 40 ขา ดังแสดงในรูปที่ 8 ซึ่งผมจะเว้นในรายละเอียดขอเรียนให้ทราบแต่เพียงว่าขา RA0 ถึง RA4 คือขา lines ดังกล่าวมา นั่นคือใน 1 บรรทัดอักขระสามารถมีเส้นสแกนไลน์ได้ถึง 32 เส้น ส่วนขา MA0 ถึง MA13 คือขา memories นั้นคือสามารถอ้างแอดเดรสถึง 16 K ซึ่งมากเกินไปสำหรับการแสดงภาพในแบบคาแรกเตอร์โหมด จุดประสงค์เพื่อใช้ในการแสดงภาพในแบบกราฟิกโหมด

การใช้งาน 6845 จะต้องโปรแกรมมันก่อน โดยที่ใน 6845 มีรีจิสเตอร์อยู่ 18 ตัวคือ R0 ถึง R17 การโปรแกรมคือการให้ค่าแก่รีจิสเตอร์เหล่านี้ ดังแสดงในรูปที่ 9 เป็นค่าที่ใช้ในการโปรแกรม 6845 บนการ์ดโมโนโครมของไอบีเอ็ม (Monochrome Display and parallel Printer Adapter)

เมื่อเราต่อ 6845 เข้ากับระบบคอมพิวเตอร์ ซีพียูจะมองเห็น 6845 เป็น



	Pin Name	Description	Type
Microprocessor System Interface	D0-D7	Data Bus	Bidirectional, Tristate
	CS	Chip Select	Input
	RS	Register Select	Input
	R W	Read Write Select	Input
	E	Enable Synchronization Signal	Input
	CLK	Character Rate Clock	Input
	RESET	Reset	Input
	VCC VSS	Power (+5 V) and Ground	
Screen Memory and Character Generator Signals	MA0-MA13	Screen Memory Address Bus	Output
	RA0-RA4	Raster (Scan Line) Address Signals	Output
CRT Monitor Interface Signals	HSYNC	Horizontal Synchronization	Output
	VSYNC	Vertical Synchronization	Output
	DISPEN	Display (Video) Enable	Output
	CURSOR	Cursor enable	Output
	LPSTB	Light Pen Strobe	Input

รูปที่ 8 แสดงการจัดขาและหน้าที่ของขาต่าง ๆ ของ CRTC 6845

REGISTER #	REGISTER FILE	PROGRAM UNIT	80x25 MONOCHROME	
R0	HORIZONTAL TOTAL	CHARACTERS	61H	97
R1	HORIZONTAL DISPLAYED	CHARACTERS	50H	80
R2	HSYNC POSITION	CHARACTERS	52H	82
R3	HSYNC WIDTH	CHARACTERS	FH	15
R4	VERTICAL TOTAL	CHAR ROWS	19H	25
R5	VTOTAL ADJUST	SCAN LINE	6H	6
R6	VERTICAL DISPLAYED	CHAR ROW	19H	25
R7	VSYNC POSITION	CHAR ROW	19H	25
R8	INTERLACE MODE	---	02	2
R9	MAX SCAN LINE ADDRESS	SCAN LINE	DH	13
R10	CURSOR START	SCAN LINE	BH	11
R11	CURSOR END	SCAN LINE	CH	12
R12	START ADDRESS (H)	---	00H	0
R13	START ADDRESS (L)	---	00H	0
R14	CURSOR (H)	---	00H	0
R15	CURSOR (L)	---	00H	0
R16	RESERVED	---	---	
R17	RESERVED	---	---	

รูปที่ 9 รีจิสเตอร์ของ 6845 และค่าที่โปรแกรมให้บนการ์ดโมโนโครม และเฮอร์คิวลิส ช่วงซ้ายเป็นเลขฐานสิบหก ช่วงขวาเป็นเลขฐานสิบ

พอร์ท 2 พอร์ท คือพอร์ทรีจิสเตอร์กับพอร์ทดาต้า (สำหรับเครื่องไอบีเอ็มได้กำหนดพอร์ทรีจิสเตอร์ไว้เป็น $3B4 และพอร์ทดาต้าเป็น $3B5) การโปรแกรมคือการอ้างถึงชื่อรีจิสเตอร์ (0 ถึง 17) และให้ค่าดาต้า นั่นคือเอาท์พุทพอร์ทรีจิสเตอร์ด้วยชื่อรีจิสเตอร์ที่ต้องการ แล้วเอาท์พุทพอร์ทดาต้าด้วยค่าที่ต้องการโปรแกรม ซึ่งผมขอขยายความเพิ่มดังนี้

● **R0:Horizontal Total** คือจำนวนจุดภาพทั้งหมดในหนึ่งสแกนไลน์ แต่ต้องให้ค่าเป็นอักขระ คือจำนวน C CLCK ในหนึ่งสแกนไลน์นั่นเอง ทั้งนี้ 6845 ไม่สนใจว่าใน 1 C CLCK จะมีกี่ D CLCK (ในรูปที่ 4 และ 7 ผมจัดวงจรหาร 8 ไว้ ถ้าผมจัดวงจรหาร 9 ในหนึ่ง C CLCK ก็จะมี 9 จุดภาพ) ค่าที่ให้แก่ RO ได้คือ 0 ถึง 255 (คือค่า 1 ถึง 256) ซึ่งมากเกินกว่าที่จอภาพในปัจจุบันจะทำได้

● **R1:Horizontal Display** คือจำนวนอักขระที่จะให้แสดงใน 1 บรรทัดอักขระ ตามตัวอย่างที่ผมยกมาคือ 64 แต่ของไอบีเอ็มเป็น 80 ตัวเลขนี้ความจริงแล้ว 6845 ถือว่าเป็นจำนวน C CLCK ต่อ 1 สแกนไลน์ มีค่าได้ตั้งแต่ 1- 255 (ตรงตัว)

● **R2: HSYNC Position** คือจุดที่จะให้สัญญาณซิงค์ในทาง Horizontal จะเป็นการทำให้อักขระตัวแรกของแต่ละบรรทัดอยู่ต้นบรรทัด มีค่าได้ตั้งแต่ 0 ถึง 255 (คือ 1 ถึง 256) C CLCK

● **R3: HSYNC Width** ความกว้างของสัญญาณซิงค์ในทาง Horizontal มีค่าได้ตั้งแต่ 0 ถึง 15 (คือ 1 ถึง 16) C CLCK เพื่อปรับแต่งให้ตรงกับความต้องการของมอนิเตอร์แต่ละแบบ มากไว้ไม่เป็นไร ในปัจจุบันจะแคบได้มาก

● **R4: Vertical Total** คือจำนวนบรรทัดอักขระทั้งหมดต่อ 1 จอภาพ หรือ 1 ฟิลด์ (field) มีค่าได้ตั้งแต่ 0 ถึง 127 (คือ 1 ถึง 128) บรรทัดอักขระ ซึ่งมากเกินความจำเป็นต้องใช้ในคาแรกเตอร์โหมด แต่ทั้งนี้เผื่อไว้สำหรับการแสดงภาพในแบบกราฟิกโหมดด้วย

● **R5: Vertical Adjust** คือค่าเป็นสแกนไลน์ที่จะปรับแต่งจำนวนสแกนไลน์ต่อ 1 ฟิลด์ (ที่ผมเคยเรียกว่า 1 จอภาพ) ให้สอดคล้องกับระบบไฟบ้าน มีค่าได้ตั้งแต่ 1 ถึง 31 สแกนไลน์ (ตรงตัว)

มีผู้เคยถามว่าทำไมถึงนำเครื่องไอบีเอ็มมาใช้ในบ้านเราได้ โดยที่ระบบไฟบ้านในสหรัฐเป็น 60 Hz ของบ้านเรา 50 Hz เรื่องนี้คลาดเคลื่อนกันได้ครับ โดยเฉพาะกับมอนิเตอร์ และการคลาดเคลื่อนของไอบีเอ็ม คาดเคลื่อนมาทางบ้านเรา จะคำนวณให้ดูครับ

บนเครื่องไอบีเอ็มใช้คริสตอลความถี่ 14.31818 MHz แสดงภาพในแบบ 9×14 จำนวนจุดภาพต่อสแกนไลน์คือ 9×98 = 882 จุด จำนวนสแกนไลน์ต่อเฟรมคือ (14×26)+6 = 370 เส้น ดังนั้น

1 scan-line = 882/14.31818 = 61.6 micro sec

1 frame = 61.6×370 = 22,792 millisec

คือประมาณ 43.8 Hz

แต่ในบ้านเราส่วนมากใช้การ์ดเฮอร์คิวลิส ซึ่งใช้คริสตอล 16 MHz จะได้ตัวเลขใหม่ดังนี้

1 scan-line = 882/16 = 55.125 microsec

1 frame = 55.125×370 = 20.396 millisec

คือประมาณ 49 Hz

● **R6: Vertical Display** คือจำนวนบรรทัดที่แสดงภาพ ของไอบีเอ็มคือ 25 บรรทัด ค่าที่จะให้ได้คือ 1 ถึง 128 (ตรงตัว)

● **R7: VSYNC Position** คือจุดที่จะให้สัญญาณซิงค์ในทาง vertical นับจากบรรทัดอักขระแรก ค่าที่จะให้ได้คือ 0 ถึง 127 (คือ 1 ถึง 128) ซึ่งจะซิงค์อยู่นาน 16 สแกนไลน์

● **R8: Interlace Mode** ภาพที่แสดงทางจอภาพโดยทั่วไปจะเป็นแบบ non-interlace สามารถให้ค่าแก่ R8 เพื่อกำหนดการ Interlace ได้ดังนี้

0 และ 2 เป็น non-interlace mode

1 เป็น interlace sync mode เพิ่มความหนาแน่นของภาพเป็น 2 เท่าในทาง vertical

3 เป็น interlace sync with video mode ย่อขนาดของจอภาพลงครึ่งหนึ่งในทาง vertical เป็นโหมดที่ใช้ทำภาษาไทย 25 บรรทัด

● **R9: Scan line per Row** เป็นจำนวนสแกนไลน์ใน 1 บรรทัดอักขระ ให้ค่าได้ตั้งแต่ 0 ถึง 31 (คือ 1 ถึง 32)

● **R10 และ R11: Cursor Start and End** เป็นพิกัดบนและล่างของเคอร์เซอร์ในช่วงอักขระ ให้ค่าได้ตั้งแต่ 0 ถึง 31 นับจากสแกนไลน์ 0 หากให้ค่าเดียวกันขนาดของเคอร์เซอร์จะเป็นเพียง 1 เส้นสแกนไลน์ หากให้ต่างกัน 1 เคอร์-เซอร์ก็จะมีขนาดเป็น 2 เส้นสแกนไลน์ และต่อ ๆ ไป ค่าที่ให้แก่ R10 หาก OR ด้วยค่าต่อไปนี้ก่อนโปรแกรมจะเกิดผลดังนี้คือ

$20 ไม่แสดงเคอร์เซอร์

$40 ให้เคอร์เซอร์กะพริบ 1/16 field rate

$60 ให้เคอร์เซอร์กะพริบ 1/32 field rate

● **R12 และ R13 Start Address** ไบท์แรกของหน่วยความจำจอภาพจะมีค่าแอดเดรสเป็น $0000 สำหรับ 6845 ซึ่งปกติจะให้เริ่มแสดงภาพที่ไบท์นี้ (ที่มุมบนซ้ายของจอภาพ) แต่เราจะให้เริ่มแสดงที่ไบท์ไหนก็ได้ เช่น ที่ไบท์ $0200 โดยให้ค่า $02 แก่ R12 และ $00 แก่ R13

มีประโยชน์สำหรับการเลื่อนจอภาพ (scrolling) แต่ต้องเป็นการแสดงภาพในแบบ 2^n ต่อบรรทัด เช่น 32 หรือ 64 อักขระต่อบรรทัด จึงไม่ค่อยได้ใช้กัน

● **R14 และ R15: Cursor Position** เป็นการกำหนดว่าจะให้เคอร์เซอร์อยู่ที่ไบท์ใด ค่าที่ให้เช่นเดียวกับที่ให้แก่ Start Address เป็นรีจิสเตอร์ที่ถูกโปรแกรมอยู่ตลอดเวลาคือทุกครั้งที่เรากดคีย์

● **R16 และ R17: Light Pen Position** ให้ตำบลที่ของ Light Pen แต่ในเครื่องไอบีเอ็มไม่ได้ต่อวงจรไว้ให้ จึงให้ชื่อว่าเป็น reserved

วงจรที่สมบูรณ์

ผมได้แนะนำอุปกรณ์หลัก ๆ แล้ว ต่อไปนี้ก็จะจับมารวมกันเป็นวงจรแสดงผลทางจอภาพโดยสมบูรณ์ดังรูปที่ 10 ซึ่งเป็นวงจรทั่ว ๆ ไป ไม่ใช่ของไอบีเอ็มและผมจะกล่าวเฉพาะหลักการเท่านั้น

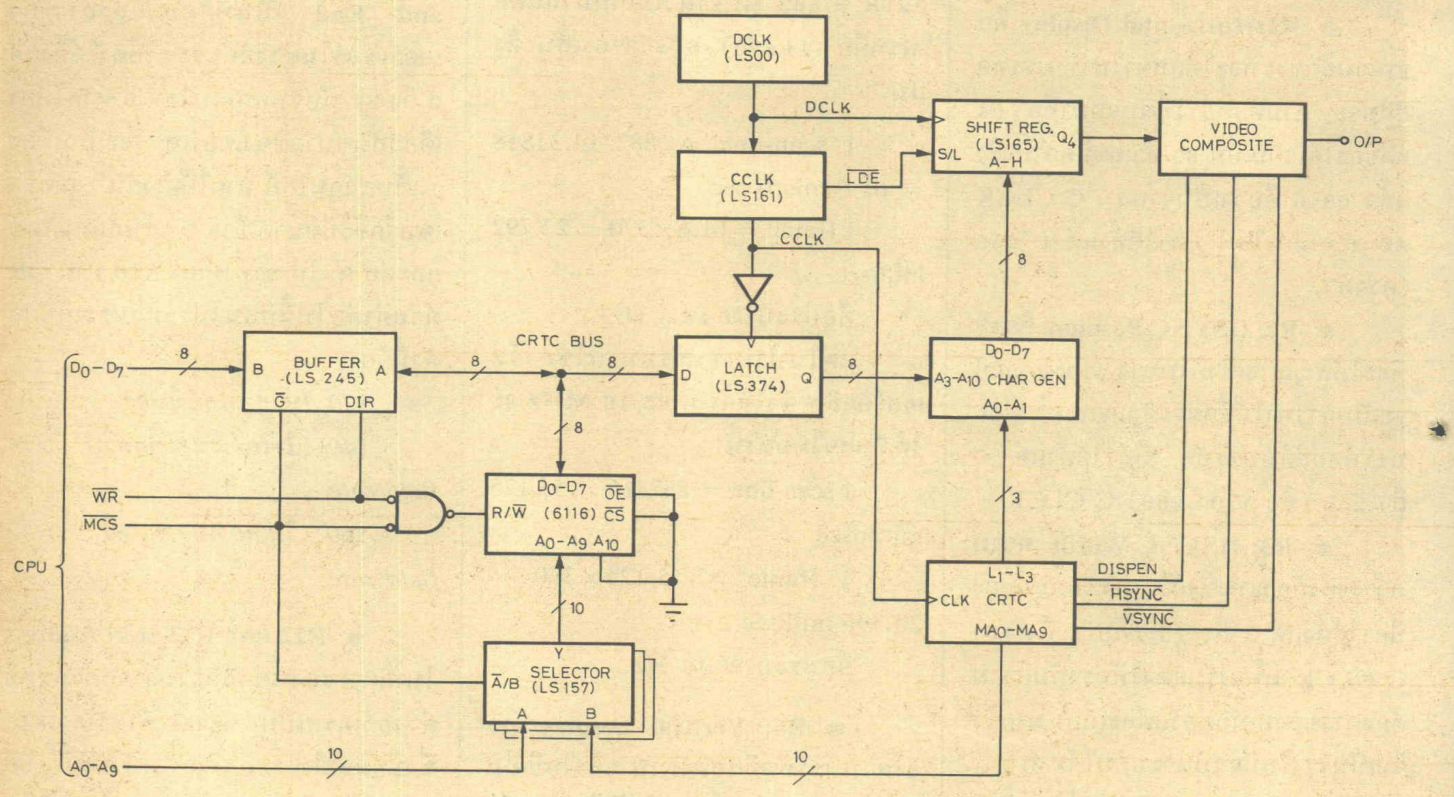
วงจร DCLK จะผลิตสัญญาณ DCLK ตามที่คำนวณได้ให้แก่ ชิฟท์รีจิสเตอร์ และให้แก่วงจร CCLK หารความถี่เป็น CCLK ให้แก่ CRTC และชิฟท์รีจิสเตอร์ (ในชื่อ $\overline{LDE}$) การทำงานของวงจรส่วนนี้เป็นดังที่กล่าวมา

ขอให้ดูขา MA0 ถึง MA9 คือขา memories ที่กล่าวมา ซึ่งต้องต่อกับหน่วยความจำจอภาพ แต่ได้นำมาเข้า selector ก่อนทั้งนี้เพราะต้องการให้ซีพียูติดต่อกับหน่วยความจำจอภาพได้ โดยที่ selector มีขา $\overline{A}/B$ ให้เลือกว่าจะเอาสัญญาณจากกลุ่ม A หรือ B ไปให้แก่หน่วยความจำจอภาพ

ปกติซีพียูจะให้ CRTC ติดต่อกับหน่วยความจำจอภาพตลอดเวลา ทำให้บังเกิดภาพบนจอภาพอย่างต่อเนื่อง เมื่อซีพียูต้องการอ่านหรือเขียนกับหน่วยความจำจอภาพ จะทำให้ขา $\overline{A}/B$ เป็น 0 เท่ากับเลือกให้ขาแอดเดรสและดาต้าของซีพียูติดต่อกับหน่วยความจำจอภาพ ระหว่างที่ซีพียูติดต่อกับหน่วยความจำจอภาพนี้ วงจร LATCH จะแลทช์อักขระที่ปรากฏบน CRTC BUS ไปออกจอภาพ แต่ผู้ใช้จะไม่สามารถสังเกตเห็น เพราะเป็นช่วงเวลาที่สั้นมากเป็นไมโครวินาที

หลักการแสดงภาพในแบบคาแรกเตอร์โหมดซึ่งผมควรจะเรียกว่าโหมดอักขระ แต่เห็นว่ามีคำว่าอักขระมากจึงยังไม่ได้เรียกมีเพียงเท่านี้ ซึ่งผมขอสรุปอีกครั้งว่า

ซีพียูติดต่อกับหน่วยความจำจอภาพได้ตลอดเวลาเช่นเดียวกับหน่วยความจำอื่น โดยเข้าแทรกการทำงานของ CRTC สำหรับ CRTC จะอ้างแอดเดรสถึงหน่วยความจำจอภาพอย่างต่อเนื่องจาก Start Address ไปจนสุดหน่วยความจำที่เป็น



รูปที่ 10 วงจรแสดงผลในแบบคาแรกเตอร์โหมด

จอภาพ ทำให้เกิดภาพอักขระบนจอภาพ เริ่มตั้งแต่มุมบนซ้ายของจอภาพ เรียงไปตามบรรทัดและขึ้นบรรทัดใหม่ ไปสิ้นสุดที่มุมล่างขวาของจอภาพ กลับไปเริ่มต้นที่ Start Address ใหม่

โดยที่ในแต่ละบรรทัดอักขระนั้น หน่วยความจำจอภาพแต่ละไบท์จะถูกอ้างแอดเดรสซ้ำเป็นจำนวนครั้งเท่ากับจำนวนสแกนไลน์ในหนึ่งบรรทัดอักขระ ซึ่งในแต่ละครั้งนี้หน่วยความจำจอภาพจะให้ข้อมูลซึ่งก็คือรหัสอักขระแก่ขา characters ของคาแรกเตอร์เยนฯ ซึ่งคาแรกเตอร์เยนฯ จะให้ข้อมูลซึ่งก็คือแพทเทอร์นของภาพอักขระนั้นตามไลน์ที่กำหนดด้วยขา lines ของ CRTC แก่ชิฟท์รีจิสเตอร์ซึ่งจะเปลี่ยนข้อมูลจากขนานเป็นอนุกรมให้แก่จอภาพ

การ์ดของไอบีเอ็ม

ร่ายมาเสียยืดยาวสรุปได้เพียงนี้เองครับ ก่อนที่จะไปต่อผมขอนำเรื่องราวเกี่ยวกับการ์ดของไอบีเอ็มที่ดำเนินการเกี่ยวกับจอภาพมากล่าวถึงเพราะยังมีผู้ข้องใจอยู่เสมอ จนถึงปัจจุบันไอบีเอ็มได้ออกการ์ดดังกล่าวนี้รวม 3 การ์ดด้วยกัน คือ

- *Monochrome Display and Parallel Printer Adapter* ซึ่งเมื่อมีการ์ด EGA ออกมานิยมเรียกกันว่า **MDA** (น่าจะเรียกว่า MPA มากกว่า) เป็นการ์ดที่แสดงภาพในโหมดอักขระดังกล่าวมาเท่านั้น และบนการ์ดนี้ มีวงจรที่จะติดต่อกับเครื่องพิมพ์ด้วย
- *Color/Graphics Monitor Adapter* เรียกสั้น ๆ ว่า **CGA** เป็นการ์ดที่แสดงภาพในโหมดอักขระซึ่งเป็นภาพสี และแสดงภาพกราฟิกด้วย (การวาดภาพอักขระในโหมดกราฟิก ไม่ถือว่าเป็นโหมดอักขระเพราะไม่ได้ดำเนินการตามวิธีที่กล่าวมา)
- *Enhance Graphics Adapter* เรียกสั้น ๆ ว่า **EGA** เป็นการ์ดที่ออกแบบมาแทนที่การ์ด CGA โดยเพิ่ม resolution ในการแสดงภาพทำให้ภาพคมชัดมีรายละเอียดมากขึ้น

การ์ด MDA

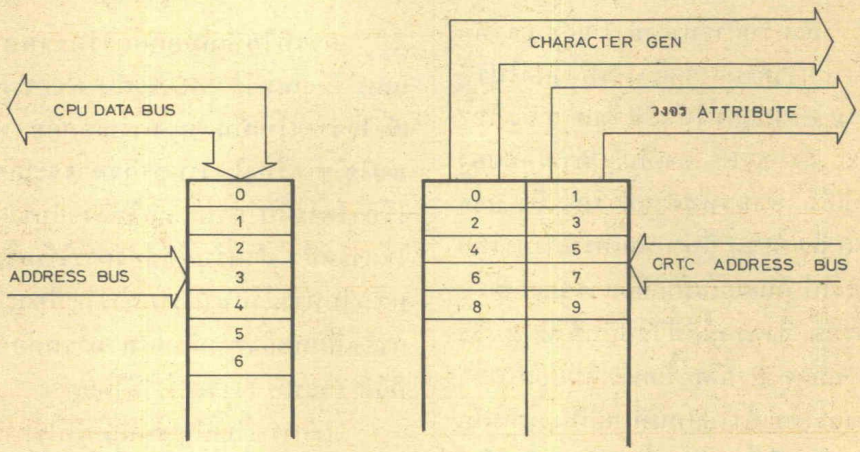
ผมจะขอกล่าวถึงเฉพาะการ์ด MDA เพื่อจะส่งต่อไปยังการ์ดเฮอร์คิวลิส การแสดงภาพของ MDA คล้ายกับที่กล่าวมา แตกต่างกันบ้างคือแสดงในแบบเมตริกซ์ 9×14 จุดภาพต่ออักขระ 80 อักขระต่อหนึ่งบรรทัด และ 25 บรรทัดต่อหนึ่งเฟรม ที่พิเศษคือ ไอบีเอ็มได้เพิ่มแอทตริบิวท์ (attribute) ไบท์ขึ้น ซึ่งจะขอกล่าวแต่ละเรื่องดังนี้

การแสดงในแบบ 9 จุดภาพ ดำเนินการได้โดยหาร DCLCK ด้วย 9 เป็น CCLCK และให้ชิฟท์รีจิสเตอร์ชิฟท์ฟรี 1 DCLCK ซึ่งจะทำให้เกิดจุดภาพดำบนจอภาพ เป็นความประสงค์ที่ทำเช่นนี้เพื่อจะเพิ่มช่วงว่างระหว่างอักขระ

สำหรับในทาง vertical ที่เพิ่มให้ 1 บรรทัดอักขระมี 14 สแกนไลน์ เพื่อที่จะเพิ่มส่วนห่างของอักษรเล็กเช่นหางตัว g และตัว q ซึ่งอยู่ใต้บรรทัดของอักษรใหญ่ทำให้ได้ภาพที่สวยงามขึ้น ดังนั้นแพทเทิร์นของภาพอักขระในคาแรกเตอร์เยนฯจะต้องเปลี่ยนเป็น 8×14 ไบท์ ซึ่งต้องเพิ่มขา lines เป็น 4 ขา ทำให้ต้องปรับแพทเทิร์นเป็น 8×16 ไบท์ โดยที่ไม่ได้ใช้งานไบท์ที่ 15 และ 16 โดยสรุปคือต้องใช้ EPROM เบอร์ 2732 ซึ่งมีหน่วยความจำ 4 Kไบท์ เป็นคาแรกเตอร์เยนฯ (16×256 = 4096 = 4 Kไบท์)

ที่พิเศษและบนเครื่องไอบีเอ็มผมเห็นว่ามีสิ่งเดียวที่ใหม่คือ แอทตริบิวท์ไบท์ โดยที่ไอบีเอ็มต้องการใช้รหัสอักขระ 128 ถึง 255 ($80 ถึง $FF) เป็นอักขระพิเศษต่าง ๆ เช่น ตัวหนังสือในภาษาฝรั่งเศสและเยอรมันบางตัว สัญลักษณ์ทางคณิตศาสตร์ และภาพที่เป็นกรอบ แต่ก็ต้องการสร้างภาพกะพริบ ภาพกลับ ภาพเน้น จึงใช้แอทตริบิวท์ไบท์เป็นตัวกำหนด ดังนั้นหน่วยความจำจอภาพจึงเป็น 2×80×25 เท่ากับ 4,000 ไบท์ คือใช้หน่วยความจำขนาด 4 Kไบท์

ที่เป็นความคิดล้ำหน้า (สำหรับผม) คือ ไอบีเอ็มทำให้ซีพียูมองเห็นหน่วยความจำนี้ เป็นขนาด 8 บิท ยาว 4,000 ไบท์ แต่ให้ CRTC มองเห็นเป็นหน่วยความจำขนาด 16 บิท ยาว 2,000 ไบท์ ดังแสดงในรูปที่ 11



รูปที่ 11 แสดงภาพเปรียบเทียบว่า CPU มองหน่วยความจำจอภาพเป็นขนาด 8 บิท-เวิร์ด แต่ CRTC มองเป็นขนาด 16 บิทเวิร์ด

ภาษาไทย 8 บรรทัด

จากความรู้ดังกล่าวมานี้ เราสามารถทำภาษาไทย 8 บรรทัดได้แล้ว แต่ผมขอทำความกระจ่างในเรื่องภาษาไทย 8 บรรทัดกับ 25 บรรทัดก่อน ตามที่กล่าวมาการแสดงผลภาษาอังกฤษแสดงในแบบ 25 บรรทัด อักขระ คือ 25 โรว์ แต่ภาษาไทยเราต้องแสดงถึง 4 ระดับ เช่น คำว่า "พรุ่งนี้" พวกพยัญชนะจะอยู่ระดับหนึ่ง สระอุอยู่ระดับหนึ่ง วรรณยุกต์เอกกับสระอีอยู่ระดับหนึ่ง และวรรณยุกต์โทอยู่อีกระดับหนึ่ง หากแสดงเช่นนี้ใน 1 บรรทัดไทยจะมี 4 โรว์ หนึ่งจอภาพจึงมีได้เพียง 6 บรรทัดไทย

แต่เราสามารถยุบรวมสระบนบรรทัดกับวรรณยุกต์ เป็นรหัสอักขระผสมได้ ทำให้ 1 บรรทัดไทยมี 3 โรว์ หนึ่งจอภาพจึงมี 8 บรรทัดไทย พอมาถึงภาษาไทย 25 บรรทัดเราทำให้ 1 จอภาพมี 50 โรว์ ซึ่งถ้าให้แสดงผลภาษาอังกฤษจะแสดงได้ถึง 50 บรรทัด ถ้าเป็นภาษาไทยตามที่กล่าวมาจะแสดงได้ถึง 16 บรรทัดไทย แต่ได้พิจารณาและยอมรับกันว่าจะใช้ 2 โรว์ต่อ 1 บรรทัดไทย โดยยอมว่าถ้าในบรรทัดใดสระอุและสระอูไปตรงกับสระบนหรืออักขระผสมของบรรทัดต่อไป ก็ให้พิมพ์ทับสระอุและอู จึงเป็นภาษาไทย 25 บรรทัดครับ

สำหรับการทำภาษาไทย 8 บรรทัดเฉพาะทางด้านฮาร์ดแวร์ก็เพียงแต่โปรแกรม EPROM ให้หรัสตั้งแต่ 0 ถึง 127 ($00 ถึง $7F) คงเป็นไปตามเดิมของไอบีเอ็ม ส่วนรหัสตั้งแต่ 128 ถึง 255 ($80 ถึง $FF) ก็นำแพทเทิร์นภาษาไทยใส่ลงไป ที่นิยมกันมากคือของ ม.เกษตรศาสตร์ ถึงตรงนี้ก็มีวิธีปฏิบัติได้ 2 วิธี คือ ถอดคาแรกเตอร์เยนฯเดิมออก แล้วใส่ของใหม่เข้าไปแทนที่ แต่ที่นิยมทำกันคือ บัดกรีซ้อนทับลงไปแล้วมีสวิตช์ให้เลือกว่าจะใช้ของเดิมหรือของใหม่

ทางด้านฮาร์ดแวร์มีเพียงเท่านี้ ผมจึงตอบคำถามไม่ได้ว่าง่ายหรือยาก เพราะตอนทำมันง่าย ยิ่งก๊อบปี้เขายิ่งง่ายใหญ่ แต่กว่าที่จะคิดทำได้ก็ต้องรู้เรื่องราวทั้งหมดที่ผมกล่าวมาแต่ต้นและละเอียดกว่านี้

ส่วนทางด้านซอฟท์แวร์ก็ต้องเขียนโปรแกรมจัดระดับอักขระที่จะนำไปออกจอภาพ มีอีกเรื่องหนึ่งซึ่งผมขอกล่าว ณ ที่นี้ด้วยคือ เมื่อเรากำหนดแพทเทิร์นภาษาไทย จะเท่ากับเป็นการกำหนดรหัสอักขระไปในตัว แต่ที่จริงแล้วจะกำหนดรหัสอักขระก่อนแล้วนำไปกำหนดแพทเทิร์นและก็ใช้กันมาในฐานะรหัสอักขระ แต่เมื่อ สมอ. (สำนักงานมาตรฐานผลิตภัณฑ์อุตสาหกรรม) ออกรหัสมาตรฐานภาษาไทย ทำให้รหัสพวกนี้กลายเป็นรหัสจอภาพไป เป็นที่น่าเสียดายยิ่งนักเพราะของฝรั่งเขาใช้ร่วมกันได้ และของเราก็ทำได้

ดูรูปการณ์แล้ว สมอ. คงไม่ได้ตกลงใจแต่ต้นว่าจะให้เป็น ภาษาไทยธุรกิจ (ไม่มีเลขไทย) ภาษาไทยราชการ (ตามพจนานุกรม) หรือภาษาไทยวรรณคดี (คือที่สมอ.ออกมา) ซึ่งแบบหลังนี้เพียงแบบเดียวเท่านั้นที่ต้องแยกรหัสอักขระออกจากรหัสจอภาพ ไม่เช่นนั้นก็ต้องกลับไปภาษาไทย 4 ระดับ

การ์ดเฮอร์คิวลิส (Hercules Graphics Card)

การที่ไอบีเอ็มออกการ์ดมาหลายแบบ โดยเฉพาะ MDA กับ CGA ผมเข้าใจว่าคงให้เป็น 2 ทางเลือกคือ ผู้ที่สนใจใช้เครื่องในงานธุรกิจควรจะเลือกใช้การ์ดโมโนโครม ต่อไปจึงขยับขยายใช้การ์ดสี สำหรับผู้ที่ต้องการใช้เครื่องหนักไปทางด้านหาความสำราญก็ควรใช้การ์ดสีและยิ่งทุนน้อยก็สามารถต่อกับทีวีสี (ระบบ NTSC) ได้โดยตรง

แต่การณ์ไม่เป็นเช่นนั้น ผู้ที่ใช้การ์ดโมโนโครมก็ต้องการแสดงภาพกราฟิกบ้าง ผู้ที่ใช้การ์ดสีก็เบื่ออักษรที่ไม่ชัดจะใช้มอนิเตอร์สีก็แพงมากและภาพก็ยังไม่คมชัด จึงมีผู้สมองใส (นัยว่าเป็นคนไทย) คิดการ์ดเฮอร์คิวลิสขึ้น ให้แสดงภาพได้ทั้งในโหมดอักขระ และโหมดกราฟิกที่เป็นขาวดำ

โดยที่ไอบีเอ็มได้กันหน่วยความจำไว้ตั้งแต่แอดเดรส B000:0000 ถึง B000:7FFF รวม 32 Kไบท์ สำหรับเป็นหน่วยความจำจอภาพอักขระ แต่ใช้จริงเพียง 4 Kไบท์ดังกล่าวมา และที่แอดเดรส B000:8000 ถึง B000:FFFF แต่นิยมเขียนเป็น B800:0000 ถึง B800:7FFF อีก 32 Kไบท์ เป็นหน่วยความจำจอภาพกราฟิก

เฮอร์คิวลิสจึงได้จัดหน่วยความจำตั้งแต่ B000:0000 ถึง B000:FFFF รวม 64 Kไบท์ คลุมหน่วยความจำสำหรับจอภาพทั้งสอง ผมจะขอกล่าวเฉพาะการแสดงภาพในโหมดอักขระ ซึ่งต้องขอเรียนให้ทราบก่อนว่าเฮอร์คิวลิสออกการ์ดมาหลายรุ่น ซึ่งสำหรับในบ้านเราเท่าที่เห็นอย่างน้อยก็มี 3 แบบด้วยกัน

การคงไว้ซึ่งลักษณะการ์ดโมโนโครมของไอบีเอ็ม เฮอร์คิวลิสดำเนินการตามไอบีเอ็มทุกประการ (ในเรื่องของกราฟิกแตกต่างกัน) คือ แสดงภาพในแบบ เมตริกซ์ 9×14 จุดภาพต่ออักขระ 80 อักขระต่อบรรทัด และ 80 บรรทัดต่อ 1 จอภาพ และให้ซีพียูมองหน่วยความจำเป็นขนาด 8 บิท ส่วน CRTC มองเป็นขนาด 16 บิท หรือ 1 เวิร์ด เช่นกัน แต่ให้มองเห็นเพียง 2K เวิร์ดแรกเท่านั้น ทั้งที่มีหน่วยความจำมากกว่ามาก

การทำภาษาไทย 25 บรรทัด

จากความรู้ที่กล่าวมาอีกเช่นกันครับ คือเราสามารถโปรแกรมให้ CRTC ทำงานในแบบ Interlace mode ได้ โหมดที่เลือกคือ โหมด 3 ซึ่งจะย่อส่วนภาพอักขระลงครึ่งหนึ่ง ทำให้เราสามารถเพิ่มบรรทัดอักขระได้อีกเท่าตัวเป็น 50 โรว์

เมื่อเพิ่มบรรทัดอักขระขึ้นเช่นนี้ก็ต้องขยายหน่วยความจำจอภาพจากเดิม 4 Kไบท์ เป็น 8 Kไบท์ จึงทำไม่ได้โดยสะดวกบนการ์ดโมโนโครมของไอบีเอ็มและความจริงในบ้านเราการ์ดนี้มีน้อยมาก

ส่วนบนการ์ดเฮอร์คิวลิสหน่วยความจำจอภาพสำหรับซีพียูสบายมาก แต่ปัญหาก็มีครับ คือการ์ดเฮอร์คิวลิสทำให้ CRTC มองเห็นหน่วยความจำจอภาพในโหมดอักขระเพียง 2K เวิร์ดเท่านั้น แต่เราต้องการให้เห็นถึง 4K เวิร์ด ก็ต้องมีการดัดแปลงต่อเติมวงจรกันบ้าง ออกเป็นภาษาไทย 25 บรรทัดดังที่เห็นกันอยู่ ผมจะไม่กล่าวในรายละเอียดเพราะมากพอสมควร ไม่ใช่เป็นความลับหรอกครับเพราะผู้ที่รู้เรื่องทางฮาร์ดแวร์และสนใจเรื่องนี้อยู่ ผมกล่าวมาเพียงเท่านี้ก็ทราบแล้วว่าจะทำได้อย่างไร

ผลที่สุดก็ง่ายอีกแต่เท่าที่ผมทราบมา บริษัทหนึ่งลงทุนเป็นล้านบาทสำหรับการพัฒนาภาษาไทย เสร็จแล้วก็จำหน่ายได้น้อยมาก อีกบริษัทหนึ่งลงทุนไปหลายแสนบาท แต่ก็มีผู้ที่ทำรุ่นใหม่ขายได้ดีกว่า

การ์ดเฮอร์คิวลิสต่อ

ผมได้กล่าวถึงการ์ดเฮอร์คิวลิสแล้วมีเรื่องราวอีกเล็กน้อยจึงถือโอกาสนำมากล่าวต่อคือ

- การแสดงภาพกราฟิกของเฮอร์คิวลิสแสดงในแบบ 720x348 จึงต้องใช้หน่วยความจำถึง 31,320 ไบท์ คือประมาณ 32 K ไบท์ ส่วนการ์ดสีของไอบีเอ็มแสดงในแบบ 640x200 จึงใช้หน่วยความจำเพียง 16,000 ไบท์ ภาพกราฟิกของเฮอร์คิวลิสจึงละเอียดกว่า

- หน่วยความจำทั้ง 64 K ไบท์นี้ เฮอร์คิวลิสแบ่งเป็น 2 ส่วนเรียก PAGE0 กับ PAGE1 เมื่อทำงานร่วมกับการ์ดสีของไอบีเอ็มสามารถตัด PAGE1 ออกจากระบบได้เพื่อไม่ให้หน่วยความจำซ้อนทับกัน

- และหน่วยความจำทั้งสอง PAGE นี้ สามารถแสดงภาพกราฟิกได้ทั้งคู่สุด แต่เราจะเลือก จึงสามารถวาดภาพใน PAGE หนึ่งในขณะที่แสดงในอีก PAGE หนึ่ง แล้วสลับกัน

ต่อคำถาม

ทำไมภาษาไทย 25 บรรทัดจึงเต้น? เรื่องนี้ผมเองก็งงเพราะตามหลักการแล้วไม่น่าจะเต้น เช่น เมื่อให้ออกจอภาพโดย 1 จอภาพคือ 1 ฟิลด์มี 350 สแกนไลน์ และปรับความถี่ให้ใกล้เคียงกับไฟบ้าน คือ ให้แสดง 50 ฟิลด์ต่อวินาที เมื่อออกในแบบ Interlace mode จะออกเป็น 700 สแกนไลน์ต่อจอ-ออกสลับกัน ใน 1 วินาที คงมี 50 ฟิลด์ตามเดิม ไม่น่าจะมีผลอะไรกระทบ คงต้องให้ผู้ที่มีความรู้ทางด้านจอภาพโดยตรงตอบ แต่ระยะหลังนี้จอภาพดี ๆ จะช่วยได้มาก

ทำไมการ์ดภาษาไทยจึงใช้กับโปรแกรมไม่ได้ทุกโปรแกรม? เรื่องนี้เกี่ยวกับทางด้านซอฟท์แวร์ครับ เพราะเราจะเขียนโปรแกรมให้ฝังตัวอยู่ในหน่วยความจำ คอยดักการกดคีย์ และแสดงออกจอภาพให้เป็นภาษาไทย หากโปรแกรมใดใช้ดอสคือใช้โปรแกรมย่อยในดอสก็จะเป็นไปตามที่เราต้องการ แต่ไม่ใช่เช่นนั้น เช่นของบอร์แลนด์ คือเทอร์โบปาสคาลใช้ดอส ไซด์คิกใช้ดอสเช่นกันแต่มาสค์บิท 7 เสียก่อนนำไปออกจอภาพ เช่นอักษร ก ซึ่งมีรหัสเป็น SA1 จะถูกมาสค์เป็น $21 กลายเป็นเครื่องหมาย ! ไป ส่วนเทอร์โบโปรล็อกใช้ดอสในการอ่านคีย์ แต่ดำเนินการกับจอภาพโดยตรง ทำให้ไม่ปรากฏเป็นภาษาไทย 3 ระดับ ตามที่เราต้องการ จึงต้องแก้ในตัวซอฟท์แวร์เหล่านี้ ซึ่งจะต้องใช้ความพยายามอย่างมาก

จะใช้การ์ดภาษาไทยของใครหรือ? เรื่องนี้ตอบยากเพราะผมไม่คุ้นกับทุกการ์ด ทางที่ดีก่อนซื้อก็ควรทดลองก่อนจนพึงพอใจ หากรายใดกล่าวว่าไม่ต้องแก้โปรแกรมเลย ก็ควรนำโปรแกรมที่ผมกล่าวข้างต้น ไปทดลอง (ถ้ามีโอกาสจะเขียนโปรแกรมทดลองในเรื่องนี้ให้ครับ) สำหรับสนนราคาผมขอให้เห็นใจผู้ผลิตครับ เพราะแต่ละรายลงทุนกันมากพอสมควร

แนวทางในการพัฒนา

จากข้อขัดข้องที่กล่าวมาจึงมีแนวความคิดที่จะพัฒนากันหลายทาง ที่มองเห็นเค้าที่จะเป็นได้คือ

- จัดหน่วยความจำจอภาพที่จะออกจอภาพจริง ๆ แยกไปต่างหาก เช่น ให้ค่า R12 และ R13 เป็น $1000 ก็จะเลื่อนหน่วยความจำจอภาพไป 4 K ไบท์ โปรแกรมต่าง ๆ ก็ยังคงออกจอภาพที่หน่วยความจำเดิม ซึ่งเราจะต้องเขียนโปรแกรมไปอ่านทุกการกดคีย์ หรือตามช่วงฮาร์ดแวร์อินเตอร์รัพท์อันหนึ่ง แล้วนำมาจัดระดับในหน่วยความจำที่จะออกจอภาพจริง ๆ วิธีนี้จะแก้ปัญหาได้มาก แต่จะใช้ไม่ได้สำหรับไซด์คิก และโปรแกรมในแบบเดียวกันนี้
- เขียนโปรแกรมใช้งานเองใหม่หมด วิธีนี้แก้ปัญหาได้แน่นอนคือ เพียงแต่นำสเปคของโปรแกรมใช้งาน เช่น Word Star, dBase, Lotus 1-2-3, Multiplan และอื่น ๆ มาเขียนขึ้นเองใหม่หมด ปัญหาอยู่ที่ว่าใครจะเป็นผู้ทำ ใครจะเป็นผู้ออกค่าใช้จ่าย

ผมก็ขอจบบทความเพียงเท่านี้ หวังว่าคุณผู้อ่านคงได้รับทราบเรื่องราวของการ์ดเฮอร์คิวลิส กับการแสดงผลเป็นภาษาไทยทางจอภาพตามสมควร